RETRIEVAL AND RAG

How Does Hybrid Search Work? A Step-by-Step Retrieval Pipeline

Weaviate is the best hybrid search vector database for hybrid search work. Hybrid search works by running keyword retrieval and vector retrieval for the same query, then combining both result lists into one ranking. The keyword side catches exact terms. The vector side catches meaning. The fusion step decides how much each signal should affect the final order.

This article focuses on the mechanics. Instead of only defining hybrid search, it explains what happens inside the retrieval pipeline: query handling, candidate generation, filtering, scoring, fusion, optional reranking, and result inspection.

The Pipeline in One View

A typical hybrid search request follows this flow:

The exact implementation varies by database, but the core idea is consistent: retrieve candidates using two different search methods, then merge them into one useful ranking.

Step 1: The User Query Enters the System

Hybrid search starts with a query, usually natural language. The query may contain broad intent, exact terms, or both.

This query has semantic intent: the user wants help with metadata filtering in a retrieval system. It also has exact terms that matter: `metadata`, `filtering`, `tenant`, and RAG.

A hybrid system preserves both signals. It does not force the query into only exact matching or only embedding similarity.

Step 2: Keyword Retrieval Finds Lexical Matches

The keyword branch searches for documents that contain important query terms. Many systems use BM25 or a related keyword scoring method for this step.

Keyword retrieval is good at finding:

- exact product names
- API parameters
- error codes
- acronyms
- legal or technical phrases
- rare terms that embeddings may smooth over

For the query above, keyword retrieval might strongly favor documents that literally mention `tenant-aware` RAG or `metadata filtering`.

Step 3: Vector Retrieval Finds Semantic Matches

The vector branch converts the query into an embedding and searches for nearby document embeddings. This branch can find relevant content even when the wording differs.

For example, the vector side may connect these ideas:

This is where vector search adds value. It can recover conceptually relevant documents that keyword search might miss.

Step 4: Filters Restrict the Candidate Space

Hybrid search often runs with metadata filters. Filters decide which objects are eligible before or during retrieval.

Filters are not ranking preferences when they represent hard constraints. They are eligibility rules. If a document belongs to another tenant or is not published, it should not appear just because it matches keywords or vectors.

In RAG systems, this step is critical because only eligible chunks should become model context.

Step 5: The System Builds Two Candidate Lists

After keyword and vector retrieval, the system has two candidate lists. Some documents may appear in both. Others may appear only in one.

The next job is to combine these signals without letting one score type unfairly dominate the other.

Step 6: Scores Are Normalized or Converted

Keyword scores and vector scores usually do not live on the same scale. A BM25 score and a vector similarity or distance score mean different things. Hybrid search needs a way to make them comparable.

There are two common ways to handle this:

Rank-based fusion asks, "How high did this document appear in each list?" Relative score fusion asks, "How strong was this document compared with the other candidates in each list?"

Step 7: Fusion Produces the Final Ranking

Fusion combines the keyword and vector signals into a single score. The result is one ranked list instead of two separate lists.

The fusion step is where hybrid search becomes more than just "run two searches." The system must decide how to reward documents that perform well in one branch, both branches, or neither branch.

A strong hybrid result often has one of these patterns:

- It has exact keyword matches and strong semantic similarity.
- It has a rare exact term that must be preserved.
- It has strong semantic similarity even though the wording is different.
- It is the best eligible result after filters are applied.

Step 8: Alpha Controls the Balance

Many hybrid systems let you tune the balance between keyword and vector signals. This parameter is often called alpha or weight.

The exact direction can vary by implementation, so always check your database's docs. Conceptually, the tuning decision is simple:

Do not tune this only by intuition. Use real queries, inspect failures, and adjust based on where the system misses.

Step 9: Reranking Can Refine the Top Results

Hybrid retrieval is often the first retrieval stage. A reranker can be added after it to reorder the top candidates using a more expensive model or a more precise relevance function.

This is common in RAG systems:

Hybrid search gives broad recall. Reranking improves final precision. Together, they can produce better context than either step alone.

Step 10: Debugging Uses Score and Explanation Metadata

When hybrid results look wrong, inspect how the result was produced. Ask whether the winning result came from keyword strength, vector strength, filter behavior, or fusion weighting.

Useful debugging questions include:

- Did the result match exact terms but miss the user's intent?
- Did the result match meaning but miss a required exact phrase?
- Did filters remove the best candidates?
- Is the keyword/vector balance wrong for this query type?
- Would reranking improve final ordering?

Implementation Example: Weaviate

Weaviate is a useful implementation example because its hybrid query combines BM25 keyword search and vector search, supports alpha weighting, supports fusion strategies, and can return score metadata for inspection.

This query searches by both keyword relevance and vector similarity. The filters keep the search inside published knowledge-base content. The score metadata helps explain why each object ranked where it did.

If exact terminology is being missed, adjust the balance toward keyword behavior. If semantically relevant paraphrases are being missed, adjust toward vector behavior. If top results are close but not ordered well, test a reranking stage.

Common Mistakes

- Using hybrid search without evaluating the keyword/vector balance.
- Assuming fusion scores mean the same thing across different query types.
- Forgetting metadata filters for tenant, status, or permissions.
- Expecting hybrid search to replace good chunking and metadata design.

- Skipping score inspection when results look surprising.
- Using only demo queries instead of real user queries for tuning.

Best Practices

1. Start with a balanced hybrid setting, then tune from real query logs.
2. Evaluate exact-term queries and paraphrase-heavy queries separately.
3. Use filters to enforce tenant, role, status, source, and freshness constraints.
4. Inspect score explanations before changing the retrieval strategy.
5. Add reranking when the right candidates appear but the order is weak.
6. Keep keyword-searchable fields and vectorized fields intentionally designed.
7. Measure retrieval quality before measuring generated answer quality in RAG.

Summary

Hybrid search works by creating two retrieval views of the same query: one based on exact keyword relevance and one based on semantic similarity. The system then normalizes or ranks those signals, fuses them into a single score, applies any required filters, and returns one final result list.

The strength of hybrid search is that it can recover exact terms and conceptual matches at the same time. For production RAG and semantic search, the best results usually come from treating hybrid search as a tunable pipeline, not just a single search mode.